



Fast Online Optimization for Terrain-Blind Bipedal Robot Walking With a Decoupled Actuated SLIP Model

Ke Wang^{1*}, Hengyi Fei² and Petar Kormushev¹

¹Robot Intelligence Lab, Dyson School of Design Engineering, Imperial College London, London, United Kingdom, ²Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom

We present an online optimization algorithm which enables bipedal robots to blindly walk over various kinds of uneven terrains while resisting pushes. The proposed optimization algorithm performs high-level motion planning of footstep locations and center-of-mass height variations using the decoupled actuated spring-loaded inverted pendulum (aSLIP) model. The decoupled aSLIP model simplifies the original aSLIP with linear inverted pendulum (LIP) dynamics in horizontal states and spring dynamics in the vertical state. The motion planning can be formulated as a discrete-time model predictive control (MPC) problem and solved at a frequency of 1 kHz. The output of the motion planner is fed into an inverse-dynamics-based whole body controller for execution on the robot. A key result of this controller is that the feet of the robot are compliant, which further extends the robot's ability to be robust to unobserved terrain variations. We evaluate our method in simulation with the bipedal robot SLIDER. The results show that the robot can blindly walk over various uneven terrains including slopes, wave fields, and stairs. It can also resist pushes of up to 40 N for a duration of 0.1 s while walking on uneven terrains.

Keywords: bipedal robot, footstep planning, uneven terrain walking, optimization, model predictive control

1 INTRODUCTION

To make bipedal robots really suitable for many applications, it is important that they can go out of the lab and walk in the complex real world environment. Real-world environments contain various kinds of uneven terrains: slopes, stairs, and hills. Most existing controllers that allow a bipedal robot to walk over uneven terrains require predefined footstep locations or exact information about the terrain height variations (Mordatch et al., 2010; Engelsberger et al., 2015; Liu et al., 2015). However, even with most advanced sensors, there are some uncertainties on the perception of the terrain. In contrast, humans can easily walk on uneven terrains, such as outdoor environments, and without extra thought or careful planning. Therefore, it is important to have a reactive controller that is robust to unobserved uneven terrain variations.

The spring-loaded inverted pendulum (SLIP) model has become a popular model for walking and running in the legged robotic research (Raibert, 1986). Despite its simplicity, it has been proven to capture essential dynamics properties of walking and running (Geyer et al., 2006). The standard setting of a SLIP model is energy-conservative: it assumes there is no energy loss at impact. Though this assumption simplifies the control analysis, it does not resemble the reality. There is energy loss on physical systems, and robots (Ahmadi and Buehler, 2006; Robotics, 2017) designed to

OPEN ACCESS

Edited by:

Enrico Mingo Hoffman,
Pal Robotics S. L., Spain

Reviewed by:

Nicola Scianca,
Sapienza University of Rome, Italy
Pierre Fernbach,
Laboratoire d'analyse et d'architecture
Des Systèmes, France

*Correspondence:

Ke Wang
k.wang17@imperial.ac.uk

Specialty section:

This article was submitted to
Humanoid Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 10 November 2021

Accepted: 03 January 2022

Published: 18 February 2022

Citation:

Wang K, Fei H and Kormushev P
(2022) Fast Online Optimization for
Terrain-Blind Bipedal Robot Walking
With a Decoupled Actuated
SLIP Model.
Front. Robot. AI 9:812258.
doi: 10.3389/frobt.2022.812258

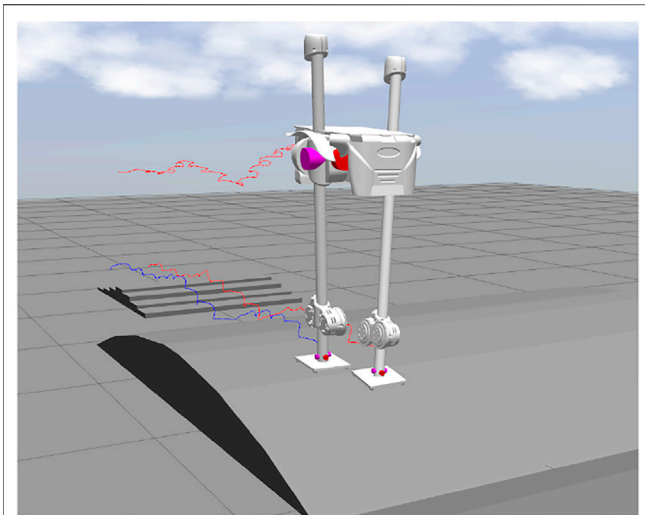


FIGURE 1 | SLIDER robot walks on the uneven terrain. The three traces of trajectories are CoM, left foot, and right foot.

approximate SLIP dynamics have added actuation to compensate for the energy dissipation. As a result, the actuated spring-loaded inverted pendulum (aSLIP) model (Ernst et al., 2010) is proposed for a better approximation of the real robot dynamics. The aSLIP model has been successfully used to design controllers not only for SLIP-like robots (Apgar et al., 2018; Xiong and Ames, 2018; Green et al., 2020) but also as a template model for humanoid robots on uneven terrain walking (Liu et al., 2016).

An important step in making a controller reactive is to achieve real-time-constrained optimization. However, due to nonlinear dynamics that arises from the 3D aSLIP model, fast optimization is difficult. Liu et al., (2016) used a gait synthesized from a library of gaits acquired from off-line optimization, but this requires a large computation load off-line and cannot cover all possible situations Apgar et al., (2018) decoupled a 3D aSLIP model to facilitate fast computation, but the continuous dynamics of the decoupled aSLIP model is still nonlinear, and there is no theoretical guarantee of fast convergence. Xin et al., (2019) used the simpler linear inverted pendulum (LIP) model to design a reactive controller for flat ground walking, but this model cannot be applied to walking on uneven terrains.

Our article proposes a reactive controller that enables robots to blindly walk over uneven terrains by optimizing horizontal footstep locations and center-of-mass (CoM) height online. “Terrain-blind” means the robot is not provided with an environment map that specifies terrain height, but the robot can still get access to its own state through state estimation, for example, the CoM height. Under the assumption that the angle the modeled inverted pendulum makes with the vertical is relatively small, we can decouple the 3D aSLIP model into a 1D-actuated spring model responsible for z direction and 2D LIP model responsible for x and y directions. The dynamics of all three dimensions can be written as linear equations in a discrete-time state space manner. We formulate the online step planner as a discrete-time model predictive control (MPC) problem and

solve it by quadratic programming (QP). To facilitate fast computation, the spring length is constrained to change linearly. As a result, we get a step planner which runs at a frequency of 1000 HZ. The step planner using a simple model is embedded into the inverse-dynamics-based whole body controller (Herzog et al., 2015; Kim et al., 2019) which tackles the inconsistency between the simple model used in high-level planning and full robot dynamics. With the whole body controller, the feet of the robot show great compliance, and this helps the robot to transit between different terrains without any information about the terrain. Due to the fast execution frequency and compliance of the foot, our proposed controller enables the robot to blindly walk over various kinds of moderately uneven terrains including slopes, wave fields, and stairs. Our controller can also handle disturbances from all direction while walking on uneven terrains. We validated our controller on the straight-legged bipedal robot SLIDER (Wang et al., 2020) in Gazebo simulation.

The main contribution of the article is the fast online optimization algorithm which enables bipedal robots to blindly walk over various uneven terrains while resisting disturbances, as shown in **Figure 1**. With reasonable assumptions, we can decouple the nonlinear 3D aSLIP dynamics and optimize the footstep location and CoM height separately. By proper reformulation, the control of the vertical dynamics can be formulated as QP, and therefore a solution is guaranteed to be found. Our controller is simple to implement and computationally efficient; the optimization of both vertical and horizontal motions can be solved by QP and executed at 1 kHz. A video showing the SLIDER robot walking with the proposed controller in simulation is available at: <https://youtu.be/ROYV-ZP8dxA>.

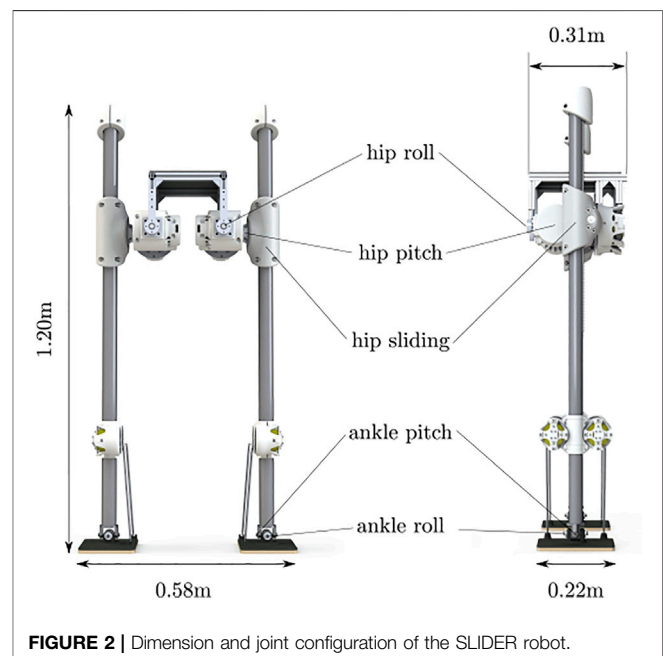
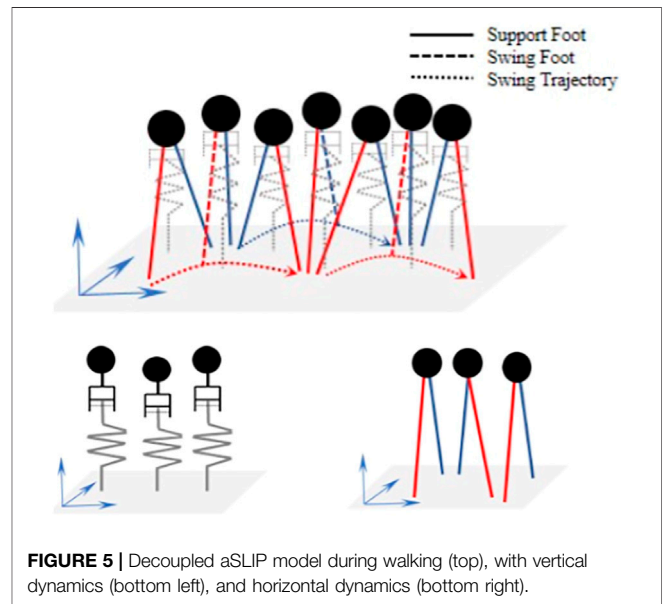
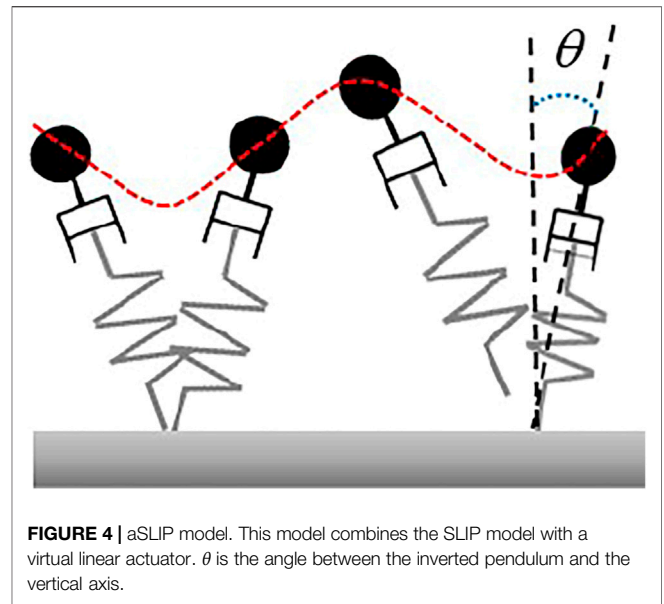
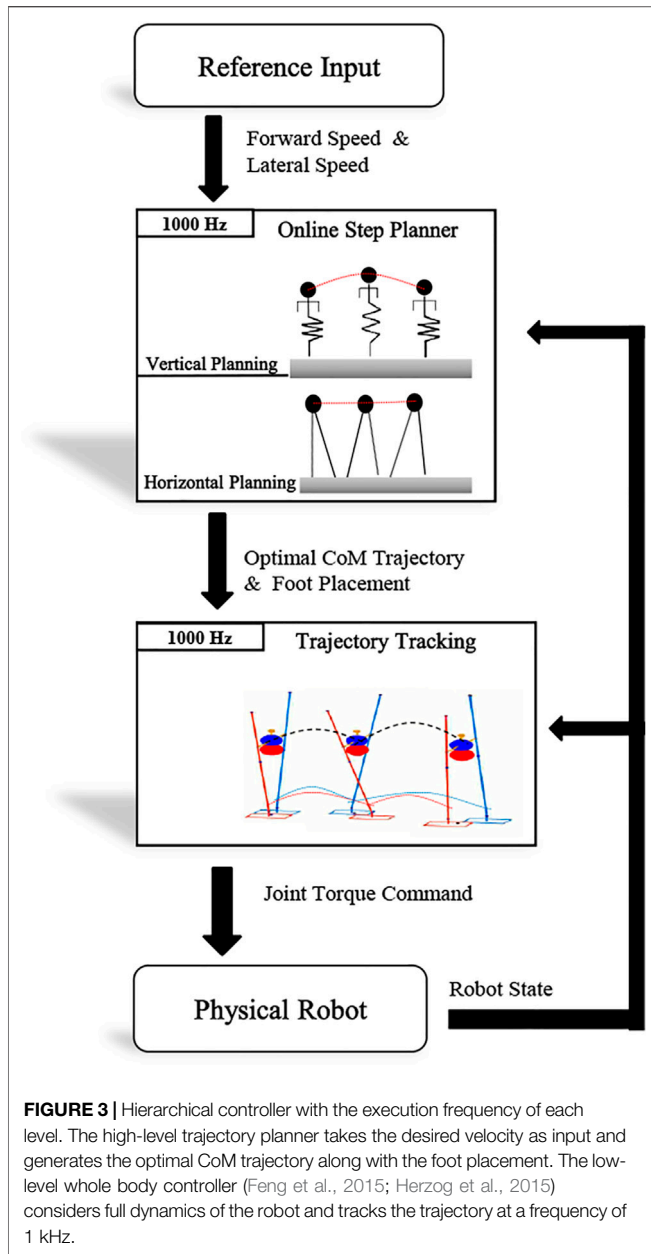


FIGURE 2 | Dimension and joint configuration of the SLIDER robot.



2 SYSTEM OVERVIEW

2.1 The SLIDER Robot

SLIDER is a knee-less bipedal robot designed by the Robot Intelligence Lab at Imperial College London, as shown in **Figure 2**. SLIDER is 1.2 m tall and has 10 degrees of freedom (DoF), namely hip pitch, hip roll, hip slide, ankle roll, and ankle pitch on each leg. The robot is very lightweight (14.5 kg in total), and most of its weight is concentrated at the pelvis. The legs are made of carbon fiber-reinforced polymer, and each leg weights only 0.4 kg. The prismatic knee joint design is a unique feature of this robot that differentiates it from many other robots with the anthropomorphic design. Due to its sliding mechanism and

lightweight leg design, SLIDER can be well approximated with an aSLIP model which can greatly simplify the planning and control problem. Moreover, the lightweight leg with a large range of motion makes the robot suitable for agile locomotion.

2.2 The Control Hierarchy

The controller presented has a hierarchical structure as shown in **Figure 3**. Due to the computational complexity of using full body motion planning, a high-level planner optimizes only the foot placement in x, y directions and the CoM height using the decoupled aSLIP model online. The low-level whole body controller (Feng et al., 2015; Herzog et al., 2015) tracks the trajectory generated by the high-level planner. The whole body

controller considers the full dynamics of the robot and generates the consistent torque command for each joint.

3 ONLINE FOOTSTEP PLANNING

Motion planning with the full dynamics of the robot is too computationally expensive to be executed at a fast frequency. Instead, we use reduced order models for online footstep planning: the LIP model is used to plan footsteps in the horizontal plane, and the 1D-actuated spring model is used to generate in the vertical direction.

3.1 The Decoupled aSLIP Model

The aSLIP model is different from the classical SLIP model in which the aSLIP model is a combination of the SLIP model and a virtual linear actuator, as shown in **Figure 4**. Introducing such a virtual actuator gives the aSLIP model the capability to actively modify the reference spring length during each walking phase and makes aSLIP model better at handling vertical changes than the SLIP model.

However, similar to the SLIP model, the dynamics of the aSLIP model presented previously is nonlinear because of the coupling in vertical and horizontal dynamics, and the exact dynamic equation needs to be numerically integrated. An approximate model, which decouples the CoM, states into horizontal and vertical directions will be used in this article. This decoupled approximate model simplifies the nonlinear dynamics of the original aSLIP model and makes the fast online optimization of footstep locations and CoM height possible. To make the approximation valid, two assumptions are made. The first assumption is that the vertical deviation of CoM over each walking period is small relative to the height of the CoM, such that individual steps can be modeled using the LIP model. The second assumption is that the angle the inverted pendulum makes with the vertical axis is small, such that the spring contributes to the CoM's vertical behavior only. This is reasonable, considering a typical small step length is small compared to the height of the CoM. These assumptions are summarized in **Figure 4**.

3.2 Vertical Dynamics of the Decoupled aSLIP Model

The original dynamics equation of the SLIP model is

$$\begin{bmatrix} m\ddot{l} \\ ml^2\ddot{\theta} \end{bmatrix} = \begin{bmatrix} ml\dot{\theta}^2 - k(l - l_0) - mg \cos \theta \\ -2ml\dot{\theta} + mgl \sin \theta \end{bmatrix}, \quad (1)$$

where l is the pendulum length and θ is the angle between the pendulum and the vertical axis, as demonstrated in **Figure 4**. Under the assumption that θ is negligible and l can be well approximated by the CoM height z , the dynamics can be simplified into the decoupled aSLIP model in vertical direction by **Eq. 2**, as shown in **Figure 5**, where r denotes the reference spring length, and k is the stiffness of the spring.

$$m\ddot{z} = -mg + k(z - r). \quad (2)$$

Compared with the classical SLIP model, the reference spring length $r(t)$ is treated as a time-varying optimization variable, which is contributed by the virtual linear actuator. We also constrain $r(t)$ to change linearly between initial and final values over the phase to facilitate fast computation. Therefore, the reference spring length should satisfy the condition below,

$$r(t) = r_0 + \frac{t}{T}(r_T - r_0), \quad (3)$$

where r_0 and r_T represent the reference spring length at the start and at the end of one step, respectively, and T is the step duration. We can formulate the dynamic **Eq. 2** into a state space equation as,

$$\dot{Z} = \begin{bmatrix} 0 & 1 \\ -\omega_z^2 & 0 \end{bmatrix} Z + \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} \left(r - \frac{g}{\omega_z^2} \right), \quad (4)$$

where the states of CoM in vertical direction are defined as $Z = [z, \dot{z}]^T$, and we define $\omega_z = \sqrt{\frac{k}{m}}$. Further, we can denote $r - \frac{g}{\omega_z^2}$ as u_z , and **Eq. 5** can be written into a linear state space equation as,

$$\dot{Z} = \begin{bmatrix} 0 & 1 \\ -\omega_z^2 & 0 \end{bmatrix} Z + \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} u_z. \quad (5)$$

We discretize **Eq. 5** with sampling time T_s and can obtain,

$$Z_{k+1} = \Phi(T_s)Z_k + \int_0^{T_s} \Phi(T_s)dt \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} u_{z,k}, \quad (6)$$

where

$$\Phi(T_s) = \begin{bmatrix} \cos(\omega_z T_s) & \sin(\omega_z T_s)/\omega_z \\ \omega_z \cdot \sin(\omega_z T_s) & \cos(\omega_z T_s) \end{bmatrix}; \quad (7)$$

$$\int_0^{T_s} \Phi(T_s)dt \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} = \begin{bmatrix} 1 - \cos(\omega_z T_s) \\ \omega_z \cdot \sin(\omega_z T_s) \end{bmatrix}.$$

We can also write this equation as:

$$Z_{k+1} = A_z(T_s)Z_k + B_z(T_s)u_{z,k},$$

where Z_k and Z_{k+1} are CoM states in the vertical direction at time k and $k+1$, respectively, $A_z(T_s)$ and $B_z(T_s)$ are time-varying matrices, and $u_{z,k}$ is the control input for the spring at time k . Therefore, N future steps can be computed with the given fixed sampling time T_s :

$$\begin{aligned} Z_1 &= A_z(T_s)Z_0 + B_z(T_s)u_{z,0} \\ Z_2 &= A_z(T_s)Z_1 + B_z(T_s)u_{z,1} \\ &\dots \\ Z_N &= A_z(T_s)Z_{N-1} + B_z(T_s)u_{z,N-1} \end{aligned} \quad (8)$$

where N is the number of steps to be optimized which should be more than 1. In this formulation, the control input vector for the spring $\bar{U} = [u_{z,0}, u_{z,1}, \dots, u_{z,N-1}]$ is the optimization variable. **Eq. 8** can also be written in a matrix form for MPC:

$$\bar{Z} = \begin{bmatrix} A_z \\ A_z^2 \\ \dots \\ A_z^N \end{bmatrix} Z_0 + \begin{bmatrix} B_z & 0 & 0 & 0 \\ A_z B_z & B_z & 0 & 0 \\ \dots & \dots & \dots & \dots \\ A_z^{N-1} B_z & A_z^{N-2} B_z & \dots & B_z \end{bmatrix} \bar{U}_z, \quad (9)$$

where \bar{Z} and \bar{U}_z are the vectors of states and control inputs in z direction, respectively.

3.3 Horizontal Dynamics of the Decoupled aSLIP Model

Under the assumption that the leg angle θ is relatively small and therefore the horizontal and the vertical dynamics can be decoupled, the horizontal dynamics then becomes a classical linear inverted pendulum (LIP) model as shown in **Eq. 10**, where z_0 is the pendulum height and remains constant within each step but may change between steps. Please note that z_0 is different from the CoM height z . Similar to Xin et al., (2019), given the continuous dynamics of LIP model in the x direction:

$$\ddot{x} = \frac{g}{z_0} (x - p_x). \quad (10)$$

The discrete state space equation can be reformulated into a linear state space as follows, with the state defined as $X = [x, \dot{x}]^T$, and input u_x defined as the footstep positions in x direction:

$$X_{k+1} = A_x(T_s)X_k + B_x(T_s)u_{x,k}, \quad (11)$$

where

$$A_x(T_s) = \begin{bmatrix} \cosh(\omega_x T_s) & \sinh(\omega_x T_s)/\omega_x \\ \omega_x \cdot \sinh(\omega_x T_s) & \cosh(\omega_x T_s) \end{bmatrix};$$

$$B_x(T_s) = \begin{bmatrix} 1 - \cosh(\omega_x T_s) \\ \omega_x \cdot \sinh(\omega_x T_s) \end{bmatrix}.$$

Similarly, states in the predicted time horizon can also be obtained by:

$$\bar{X} = \begin{bmatrix} A_x \\ A_x^2 \\ \dots \\ A_x^N \end{bmatrix} X_0 + \begin{bmatrix} B_x & 0 & 0 & 0 \\ A_x B_x & B_x & 0 & 0 \\ \dots & \dots & \dots & \dots \\ A_x^{N-1} B_x & A_x^{N-2} B_x & \dots & B_x \end{bmatrix} \bar{U}_x, \quad (12)$$

where \bar{X} and \bar{U}_x are the vectors of states and control inputs in x direction, respectively. The dynamics in the y direction has identical formulation to the dynamics in the x direction.

3.4 Foot Placement and CoM Trajectory Optimization

The footstep planning can be formulated as a QP problem due to the fact that both horizontal and vertical dynamics are linear and the constraints are linear. The formulation is:

$$\begin{aligned} \min_{u_x, u_y, u_z} \quad & \Gamma \quad (\text{costfunction}); \\ \text{s.t.} \quad & X_{k+1} = A(T_s)X_k + B(T_s)u_k \quad (\text{dynamics}); \\ & h(\mathbf{p}_j) < 0 \quad (\text{reachability}), \end{aligned}$$

where $\Gamma = \Gamma_1 + \Gamma_2 + \Gamma_3$ is the cost function term, X and u are general representations of states and control inputs, respectively, and \mathbf{p}_j is the left or right foot position.

The cost formulation is composed of three parts: Γ_1 , Γ_2 , and Γ_3 . The first part is minimizing the difference between the predicted

state and the referenced state, which drives the CoM state to reach the desired one from the given current state. The formulation is:

$$\Gamma_1 = \|X_N - X_N^{ref}\|_P^2 + \sum_{k=0}^{N-1} \|X_k - X_k^{ref}\|_Q^2, \quad (13)$$

where P and Q are the weight matrices. We only care about the velocity of CoM in all directions because we want a reactive footstep planner which is not constrained by an absolute reference trajectory. From another point of view, tracking the reference velocity is equivalent to tracking the relative CoM reference position. X^{ref} is easy to define with the desired sagittal and frontal velocity. However, the reference velocity in the vertical direction is not constant during one step. To get the reference velocity, we can integrate **Eq. 2** and get the continuous dynamics equation:

$$z(t) = d_1 \cos \omega_z t + d_2 \sin \omega_z t + r(t) - g/\omega_z^2, \quad (14)$$

where

$$d_1 = z_0 - r_0 + g/\omega_z^2; \quad (15)$$

$$d_2 = \dot{z}_0/\omega_z - (r(t) - r_0)/(T\omega). \quad (16)$$

In this equation, z_0 and \dot{z}_0 are the CoM vertical position and velocity at the start of the current step, respectively. Then, the time-varying reference velocity in z direction can be derived from the continuous dynamics.

The second part is minimizing the difference between the reference control input u_k^{ref} and the optimized control input u_k and R is the weight matrix.

$$\Gamma_2 = \sum_{k=0}^{N-1} \|u_k - u_k^{ref}\|_R^2. \quad (17)$$

For the reference control input in z direction, recall that the spring length is constrained to change linearly, as indicated by **Eq. 3**. If we denote Δr_i as the change of spring length during one sampling time T_s at step i , then it is clear that Δr_i is constant throughout step i , and the reference control input in z direction \bar{U}_z^{ref} is therefore obtained as below:

$$\bar{U}_z^{ref} = \begin{bmatrix} r_0 I_{N_r} \\ 0_{N_s} \\ \dots \\ 0_{N_s} \end{bmatrix} + \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} 0_{N_r} \\ \Delta r_1 I_{N_s} \\ \dots \\ \Delta r_{N_{steps}} I_{N_s} \end{bmatrix}, \quad (18)$$

where N_r is the remaining number of sampling points for the current step and N_s is the number of sampling points for one step. N_{steps} is the number of predicted steps.

For the horizontal trajectory planning, the input of the system is the footstep position. The reference control input in x direction \bar{U}_x^{ref} is then obtained according to the current support foot position P_0 , and the difference between two consecutive steps $\Delta P_{x,i}$.

$$\bar{U}_x^{ref} = \begin{bmatrix} I_{N_r} \\ 0_{N_s} \\ \dots \\ 0_{N_s} \end{bmatrix} P_{x,0} + \begin{bmatrix} 0_{N_r} & \dots & 0_{N_r} \\ I_{N_s} & \dots & 0_{N_s} \\ \dots & \dots & \dots \\ 0_{N_s} & \dots & I_{N_s} \end{bmatrix} \begin{bmatrix} P_{x,1} \\ P_{x,2} \\ \dots \\ P_{x,N_{steps}} \end{bmatrix}; \quad (19)$$

$$\begin{bmatrix} P_{x,1} \\ P_{x,2} \\ \dots \\ P_{x,N_{step}} \end{bmatrix} = \begin{bmatrix} I_{N_f} \\ I_{N_s} \\ \dots \\ I_{N_s} \end{bmatrix} P_{x,0} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta P_{x,1} \\ \Delta P_{x,2} \\ \dots \\ \Delta P_{x,N_{step}} \end{bmatrix}.$$

The third part is tracking the desired change of the control input between two consecutive steps, with d_i denoting the desired difference and W denoting weights. For the vertical trajectory planning, we set the desired difference equal to zero; this means the footstep planner tries to keep a constant CoM height during one step to make the decoupled aSLIP assumption valid.

$$\Gamma_3^z = \sum_{i=1}^{N_{steps}} \|\Delta r_i - d_i^z\|_W^2 = \sum_{i=1}^{N_{steps}} \|\Delta r_i\|_W^2. \quad (20)$$

For horizontal trajectory planning,

$$\Gamma_3^{x,y} = \sum_{i=1}^{N_{steps}} \|\Delta P_i^{x,y} - d_i^{x,y}\|_W^2. \quad (21)$$

In x direction, d_i^x is the step length which is calculated by the desired speed multiplied by the step time. In y direction, d_i^y is defined as $d_i^y = d_{step} * 2 * (-1)^j$, where d_{step} is the desired inter-feet clearance distance, and j is the flag for the supporting foot, 0 stands for left support, and 1 stands for right support. This relative distance regularization term is introduced to keep the feet away from each other to avoid self-collision. Moreover, as this part of the cost function only includes relative distances, it helps to produce a reactive footstep planner which keeps the robot walking even when unexpected disturbance is applied.

The reachability constraint is responsible for making sure the footstep location is physically possible,

$$\mathbf{L} - \mathbf{r}^{x,y} < \mathbf{p}_j - \mathbf{c} < \mathbf{L} + \mathbf{r}^{x,y}, \quad (22)$$

where \mathbf{L} is the nominal offset of the foot position from the CoM of the robot, $\mathbf{r}^{x,y}$ is the reachability constraint in the x and y directions, and \mathbf{c} is the CoM position.

4 TRAJECTORY TRACKING

The low-level trajectory tracking generates corresponding torques for each joint to minimize the difference between the actual body trajectory and the desired trajectory given by the high-level trajectory planner. Since the trajectory planning does not include the full body dynamics, the dynamic inconsistency of the torque command generated by the high-level planner is significant. The whole body controller solves the inverse dynamics based on the full robot dynamics.

4.1 Rigid Body Dynamics

The walking robot is modeled as a floating-based rigid body system with coordinates $q = [q_b, q_r]$. Here, $q_b \in \mathbb{R}^7$ represents the position and orientation of the floating base using quaternions, and $q_r \in \mathbb{R}^{10}$ represents the joint configuration. Inspired by Herzog et al., (2015), the full dynamics can be decomposed into an underactuated part and an actuated part:

$$\begin{bmatrix} M_f \\ H_f \\ M_a \end{bmatrix} \ddot{q} + \begin{bmatrix} H_f \\ H_a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ S_a \end{bmatrix} \tau + \begin{bmatrix} J_f^T \\ J_a^T \end{bmatrix} f, \quad (23)$$

where M , H , S_a , τ , J , and f are the mass matrix, the Coriolis force vector and the gravitation force vector, the actuator selection matrix, the joint torques vector, the stacked contact Jacobian, and the reaction force vector, respectively. The subscripts f and a indicate the floating part and the actuated part, respectively.

4.2 Contact Force Constraint

Friction is an important contributing factor to the stability of walking to prevent the foot from slipping. We use the pyramidal friction model which is a linear inequality constraint for fast optimization:

$$f_z \geq 0, \quad |f_x| \leq \frac{\mu f_z}{\sqrt{2}}, \quad |f_y| \leq \frac{\mu f_z}{\sqrt{2}}, \quad (24)$$

where μ is the friction coefficient.

4.3 Swing Foot Trajectory Generation

The swing foot trajectory is generated using a fifth-order polynomial. The start and final positions and velocities and accelerations are specified, and a parametric quintic curve is generated in x , y , and z directions. The polynomial in z direction has two halves, with the predefined foot height and zero velocity at the midpoint. The start and final positions in z direction are calculated using the estimated CoM z position at the start of current step. Because the robot is walking blindly, we do not define a trajectory for the foot orientation, and this allows the foot to be compliant to a range of surfaces.

4.4 Whole Body Controller

The whole-body controller takes the responsibility of computing the joint torques to achieve the desired motions defined in the operational space while respecting a set of constraints. The tasks of interest in this article are the CoM position, the pelvis orientation, the angular momentum of the robot, and the feet positions and orientations. The task for the linear motion can be expressed as:

$$J_T \ddot{q} = \ddot{x}^{\text{cmd}} - \dot{J}_T \dot{q}; \quad (25)$$

$$\dot{x}^{\text{cmd}} = \dot{x}^{\text{des}} + K_p^{\text{pos}}(x^{\text{des}} - x) + K_D^{\text{pos}}(\dot{x}^{\text{des}} - \dot{x}), \quad (26)$$

where J_T is the translational Jacobian for the task and x is the actual position of the link. For the task of angular momentum, the centroidal momentum matrix (Orin et al., 2013) is used as the Jacobian for the task. In uneven terrain walking, the angular momentum task is defined as a damping task that damps out excess angular momentum to make the walking robot more stable. For the task of the angular motion, the command can be formulated as:

$$J_R \ddot{q} = \dot{\omega}^{\text{cmd}} - \dot{J}_R \dot{q}; \quad (27)$$

$$\dot{\omega}^{\text{cmd}} = \dot{\omega}^{\text{des}} + K_p^{\text{ang}}(\text{AngleAxis}(R^{\text{des}} R^T)) + K_D^{\text{ang}}(\omega^{\text{des}} - \omega), \quad (28)$$

where J_R is the rotational Jacobian for the task, R and R^{des} denote the actual and desired orientation of the pelvis link, respectively,

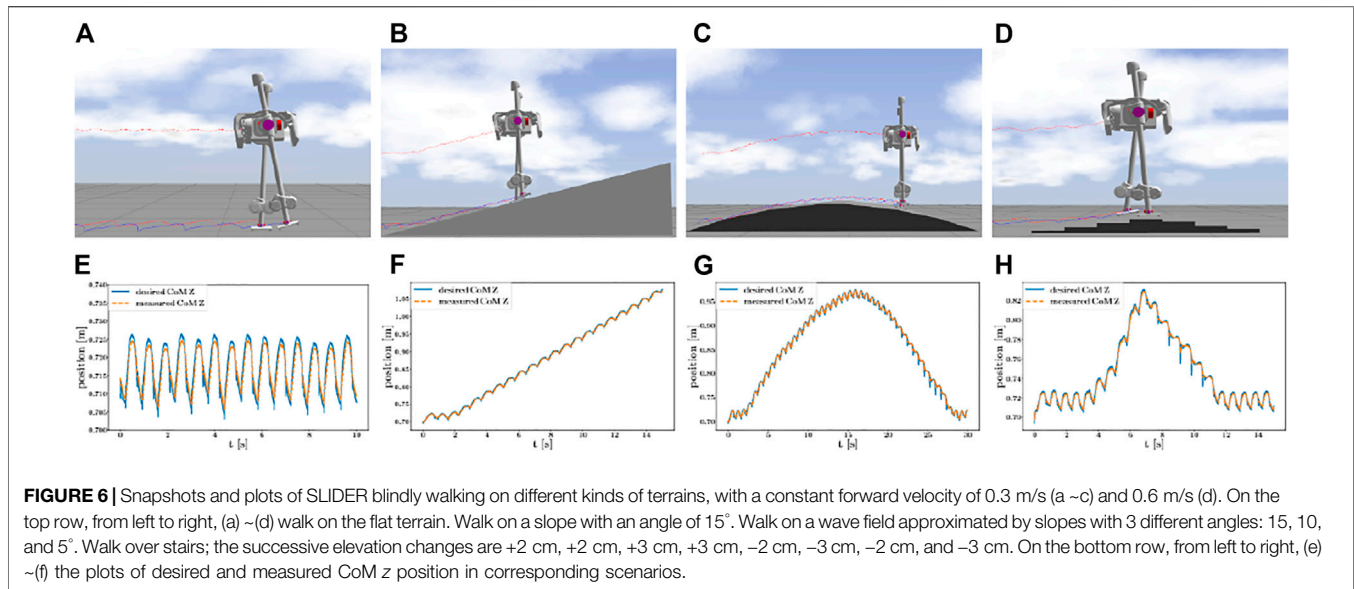


FIGURE 6 | Snapshots and plots of SLIDER blindly walking on different kinds of terrains, with a constant forward velocity of 0.3 m/s (a–c) and 0.6 m/s (d). On the top row, from left to right, (a)–(d) walk on the flat terrain. Walk on a slope with an angle of 15° . Walk on a wave field approximated by slopes with 3 different angles: 15° , 10° , and 5° . Walk over stairs; the successive elevation changes are $+2$ cm, $+2$ cm, $+3$ cm, $+3$ cm, -2 cm, -3 cm, -2 cm, and -3 cm. On the bottom row, from left to right, (e)–(h) the plots of desired and measured CoM z position in corresponding scenarios.

AngleAxis () maps a rotation matrix to the corresponding axis–angle representation, and $\omega \in \mathbb{R}^3$ is the angular velocity of the link. The angular motion is needed because when walking on the uneven terrain, the pelvis orientation needs to be regulated around a nominal orientation to maintain a good posture. We set small values for K_p^{ang} and K_d^{ang} in the foot orientation task to make the ankle compliant so that the foot can adapt to different terrains. The whole body controller can be formulated as a QP problem as follows:

$$\begin{aligned} \min_{\ddot{q}, f} \quad & \|A\ddot{q} + \dot{A}\dot{q} - X^{\text{cmd}}\|_W^2; & (29) \\ \text{s.t.} \quad & M_f \ddot{q} - J_f^T f = -H_f \quad (\text{floatingbasedynamics}); \\ & Pf \leq 0 \quad (\text{pyramidalfrictioncone}); \\ & S_a^{-1} (M_a \ddot{q} + H_a - J_a^T f) \in [\tau_{\min}, \tau_{\max}] \quad (\text{inputlimits}), \end{aligned}$$

where A is a stack of the Jacobian matrices for the tasks of interest, X^{cmd} is a stack of the commanded accelerations, and W is the weighting matrix, and P denotes the linearized friction cone. We treat the unilateral contact constraint as a soft constraint by simply assigning a large weight on the desired zero acceleration of the foot (Kuindersma et al., 2016; Apgar et al., 2018). This can speed up the optimization, and it is reported in Feng et al., (2015) that this gives better stability. The output joint torque commands τ at each control iteration can be computed by

$$\tau = S_a^{-1} (M_a \ddot{q} + H_a - J_a^T f). \quad (30)$$

5 RESULTS

This section discusses implementation details and simulation results of the SLIDER robot walking on different kinds of uneven terrains including slopes, wave fields, and stairs, as shown in Figure 6. All the experiments are included in the accompanying video <https://youtu.be/ROYv-ZP8dxA>.

5.1 Implementation

Both the high-level footstep planner and low-level trajectory tracking controller are implemented in C++ for real-time performance. In the whole body controller, we use Pinocchio (Carpentier et al., 2015–2021) to compute the full rigid body dynamics and qpOASES (Ferreau et al., 2014) to solve the related QP problem in both levels. All experiments were carried out in the robot simulation environment Gazebo (Koenig and Howard, 2004) with the physics engine ODE (Drumwright et al., 2010), using the full dynamics of the real SLIDER robot. The communication through different levels of the control hierarchy is achieved through ROS.

The decoupled aSLIP parameters are set to match with the physical SLIDER robot, here $r_0 = 0.715$ m, $m = 14.5$ kg, and $k = 1470$ N/m. The step duration is chosen to be 0.7 s, and the foot height in the swing foot trajectory generation is 5 cm. The sampling time in the discrete-time MPC is 0.1 s in x and y directions and 0.05 s in z direction. The motion planner predicts four steps in horizontal states and one step in the vertical state. We use the same parameters among all walking experiments except that the forward velocity is different. In the stair-walking experiment, the robot has to walk faster otherwise the foot might hit the edge of the stair.

5.2 Flat Ground Walking

We first validate our approach on the flat ground. Because there are no changes in the z direction on the flat ground, the z position of CoM oscillates around r_0 , as shown in Figures 6A,E.

5.3 Walking on the Smooth Uneven Terrain

We then validate our approach on slopes and wave fields where the change of terrain height is smooth. With a forward velocity of 0.3 m/s, the SLIDER robot can walk on a slope with 15° and a wave field approximated by slopes with three different angles: 15° , 10° , and 5° , as shown in Figures 6B,C,F,G. In the experiment, the

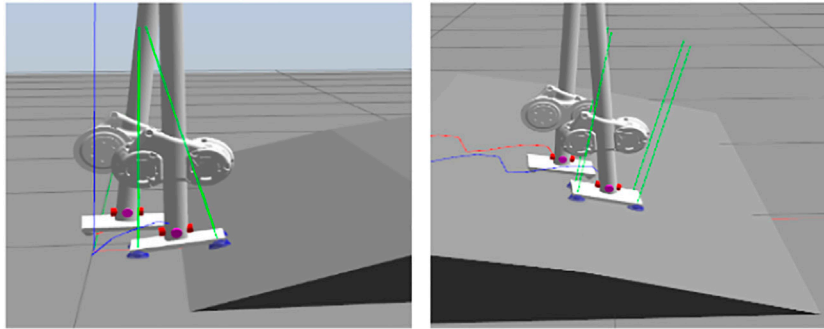


FIGURE 7 | With the whole body controller, the foot is compliant, and the robot can perform transitions to different terrains even though the robot does not know how the terrain looks like. This makes the robot robust to unobserved terrain variations. The green lines indicate the magnitude and direction of the ground reaction forces. The blue and red lines show the trace of the right and left foot centers, respectively.

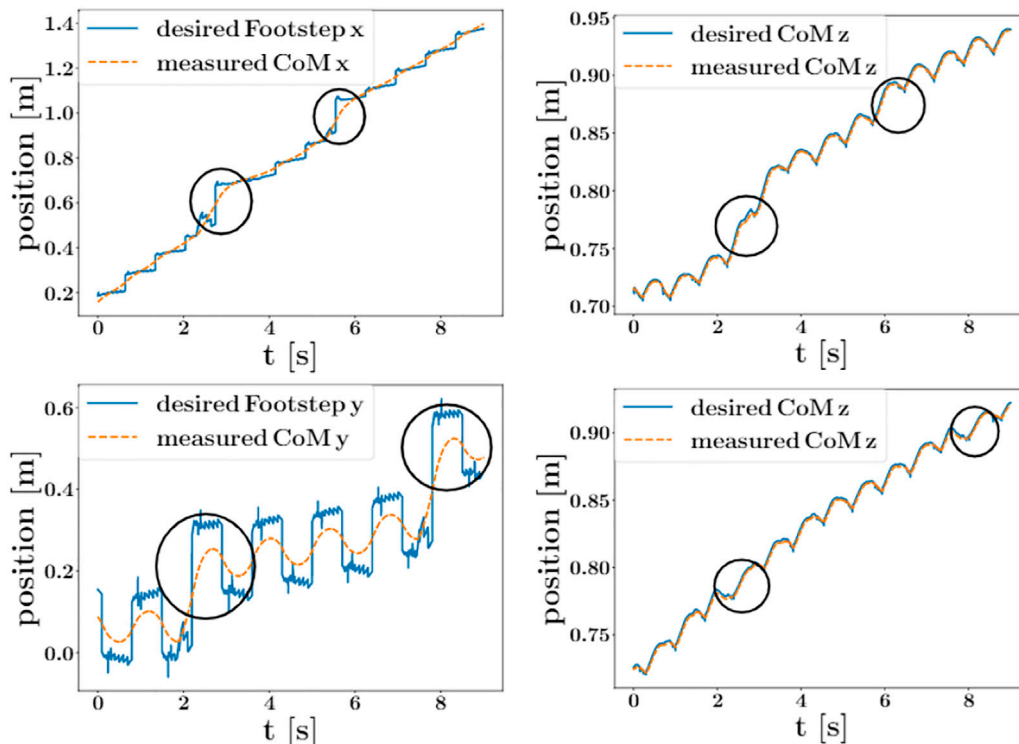


FIGURE 8 | Comparison between the proposed aSLIP planner with the LIP planner when walking on a 15° slope. Top left is the snapshot showing SLIDER walking using a LIP planner; after two steps, the robot stops walking forward because LIP does not take height change into account. Top right is the snapshot showing SLIDER walking using our aSLIP planner where the robot is able to walk forward successfully. Bottom is a plot showing the comparison between the desired CoM height of LIP and aSLIP planners.

whole body controller plays an important role in making the robot remain robust to unobserved terrain variations. By properly tuning the PD gain of the foot orientation task in the whole body controller, the foot is compliant to adapt to large terrain variations, as shown in **Figure 7**.

Further experiments, such as pushing the robot while it was walking on uneven terrains, were performed. As shown in

Figure 8, the robot was pushed in x or y direction twice while it was walking on a wave field. When the robot was pushed in x direction, the robot quickly took a large step in x direction to regulate the CoM velocity back to the desired velocity. There is also a big change in z direction because the wave field is ascending in x direction when the robot got pushed, but the motion quickly got stabilized. In the case of y direction push,

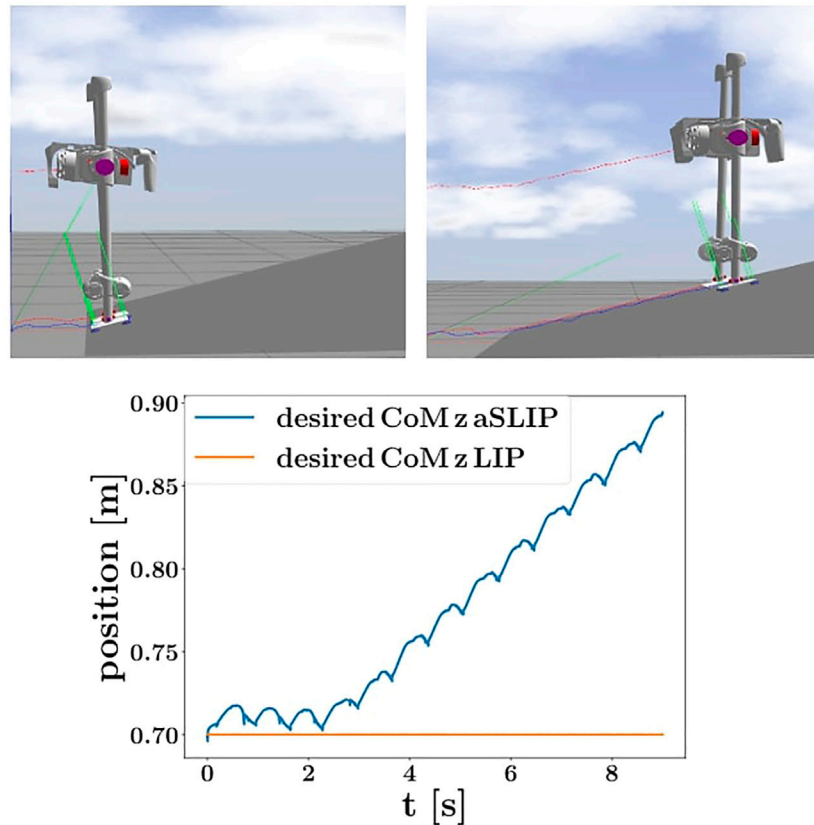


FIGURE 9 | Plots showing the states of CoM when the robot got pushed in x or y direction while walking on a wave field at a speed of 0.3 m/s. The pushes are indicated with circles. All impulses are applied with a value of 40 N for a duration of 0.1 s. Top row: states of CoM when got pushed in x direction. Bottom row: states of CoM when got pushed in y direction.

the robot also took a large step in y direction and stabilized in one step. As the terrain height is only changing along x direction, there is no big change in z direction in this case.

We also compared our proposed planner with a planner using LIP, as shown in **Figure 9**. The robot with a LIP planner is unable to walk up a 15° slope as the LIP planner assumes a constant CoM height, as shown in the bottom plot in **Figure 9**. The aSLIP planner enables SLIDER to walk uphill because it adapts to height changes.

5.4 Walking on the Discrete Uneven Terrain

The discrete uneven terrain provides bigger instantaneous variations to terrain height than the smooth uneven terrain. As shown in **Figure 6D**, the robot walked blindly over a set of stairs with a biggest height change of 3 cm. The robot walked with a forward velocity of 0.6 m/s and a foot height of 5 cm in the swing foot phase. As shown in **Figure 6H**, the CoM Z position exhibits large variations when walking over stairs but was stabilized quickly. In our experiment, the highest step the robot can walk over is 3 cm, for higher steps the foot would hit the edge of the stair and get stuck.

6 DISCUSSION

The proposed method is applied to the SLIDER robot and demonstrates successful blind walking on various uneven terrains and the robustness to disturbances due to fast MPC. Because of SLIDER's lightweight legs, the robot can perform fast leg movements which helps to stabilize the robot. Furthermore, the proposed method is general and can easily be applied to other legged robots.

There are other techniques dealing with variable height CoM trajectories, for example, DCM-based approaches (Englsberger et al., 2013; Caron and Kheddar, 2017). But these techniques require the controller to be terrain-aware to plan future CoM motion. Our proposed controller adjusts the CoM height online within each step by following a spring dynamics. This has two advantages: first, the controller does not require terrain information; second, the spring dynamics enables the vertical compliance of the robot so that the robot is robust to unexpected height variations.

The author observed that the tracking error is larger when the robot walks down the wave field or stairs than walking up, as

shown in **Figures 6G,H**. This happens because the feet are still in the air at the end of one step when the robot walks down. The unexpected sudden drop of the feet at the start of the next step gives the robot a large impact. For robot walking upward, the effect of early touchdown can be alleviated by the compliant feet and vertical compliance of the CoM. To improve, a controller with contact detection or terrain information can achieve a smaller tracking error.

7 CONCLUSION AND FUTURE WORK

We present an online optimization algorithm, which enables robots to blindly walk over various kinds of uneven terrains while resisting pushes. The high-level motion planner performs fast online optimization on footstep locations and CoM height, and the low-level inverse-dynamics-based whole body controller tracks the trajectory. We show in simulation that using this controller, the robot SLIDER can walk over slopes, wave fields, and stairs without any terrain information and can also recover from pushes while walking. Future work will involve implementing our approach on the real SLIDER robot and also incorporating perception information into the high-level step planner.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**; further inquiries can be directed to the corresponding author.

REFERENCES

- Ahmadi, M., and Buehler, M. (2006). Controlled Passive Dynamic Running Experiments with the Arl-Monopod II. *IEEE Trans. Robot.* 22, 974–986. doi:10.1109/tro.2006.878935
- Apgar, T., Clary, P., Green, K., Fern, A., and Hurst, J. W. (2018). Fast Online Trajectory Optimization for the Bipedal Robot Cassie. *Robotics: Sci. Syst.* 101, 14. doi:10.15607/rss.2018.xiv.054
- Caron, S., and Kheddar, A. (2017). “Dynamic Walking over Rough Terrains by Nonlinear Predictive Control of the Floating-Base Inverted Pendulum,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, Canada, September 24–28, 2017, 5017–5024. doi:10.1109/IROS.2017.8206385
- [Dataset] Carpentier, J., Valenza, F., and Mansard, N. (2015–2021). Pinocchio: Fast Forward and Inverse Dynamics for Poly-Articulated Systems. Available at: <https://stack-of-tasks.github.io/pinocchio> (Accessed January 16, 2022).
- Drumwright, E., Hsu, J., Koenig, N., and Shell, D. (2010). “Extending Open Dynamics Engine for Robotics Simulation,” in International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Darmstadt, Germany, November 15–18, 2010 (Springer), 38–50. doi:10.1007/978-3-642-17319-6_7
- Engelsberger, J., Ott, C., and Albu-Schaffer, A. (2015). Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion. *IEEE Trans. Robot.* 31, 355–368. doi:10.1109/tro.2015.2405592
- Engelsberger, J., Ott, C., and Albu-Schaffer, A. (2013). “Three-Dimensional Bipedal Walking Control Using Divergent Component of Motion,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3–7, 2013 (IEEE), 2600–2607. doi:10.1109/iro.2013.6696723

AUTHOR CONTRIBUTIONS

KW significantly contributed to the development of the presented approach. KW implemented the first version of the code, and HF improved the implementation. KW and HF conducted the experiments in simulation. KW made the video. KW and PK contributed to writing and reviewed the submitted version.

FUNDING

The experiments in this paper are part of the SLIDER project that has received funding from the European Union’s Horizon 2020 research and innovation programme, via the FSTP-2 Open Call issued and executed under Project EUROBENCH (grant agreement 779963).

ACKNOWLEDGMENTS

The authors would like to thank Digby Chappell for meaningful discussions. This work utilized expertise and prototyping equipment at the Imperial College Advanced Hackspace.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2022.812258/full#supplementary-material>

- Ernst, M., Geyer, H., and Blickhan, R. (2010). “Spring-Legged Locomotion on Uneven Ground: a Control Approach to Keep the Running Speed Constant,” in *Mobile Robotics: Solutions and Challenges* (World Scientific), 639–644.
- Feng, S., Whitman, E., Xinjilefu, X., and Atkeson, C. G. (2015). Optimization-Based Full Body Control for the Darpa Robotics challenge. *J. Field Robotics* 32, 293–312. doi:10.1002/rob.21559
- Ferreau, H. J., Kirches, C., Potschka, A., Bock, H. G., and Diehl, M. (2014). Qpoases: A Parametric Active-Set Algorithm for Quadratic Programming. *Math. Prog. Comp.* 6, 327–363. doi:10.1007/s12532-014-0071-1
- Geyer, H., Seyfarth, A., and Blickhan, R. (2006). Compliant Leg Behaviour Explains Basic Dynamics of Walking and Running. *Proc. R. Soc. B.* 273, 2861–2867. doi:10.1098/rspb.2006.3637
- Green, K., Hatton, R. L., and Hurst, J. (2020). “Planning for the Unexpected: Explicitly Optimizing Motions for Ground Uncertainty in Running,” in 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, May 21–August 21, 2020 (IEEE), 1445–1451. doi:10.1109/icra40945.2020.9197049
- Herzog, A., Rotella, N., Mason, S., Grimminger, F., Schaal, S., and Righetti, L. (2015). Momentum Control with Hierarchical Inverse Dynamics on a Torque-Controlled Humanoid. *Auton. Robot.* 40, 473–491. doi:10.1007/s10514-015-9476-6
- Kim, D., Di Carlo, J., Katz, B., Bledt, G., and Kim, S. (2019). Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control. arXiv preprint arXiv:1909.06586.
- Koenig, N., and Howard, A. (2004). “Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, September 28–October 4, 2004, 2149–2154.

- Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., et al. (2016). Optimization-Based Locomotion Planning, Estimation, and Control Design for the Atlas Humanoid Robot. *Auton. Robot* 40, 429–455. doi:10.1007/s10514-015-9479-3
- Liu, Y., Wensing, P. M., Orin, D. E., and Zheng, Y. F. (2015). “Trajectory Generation for Dynamic Walking in a Humanoid over Uneven Terrain Using a 3d-Actuated Dual-Slip Model,” in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, September 28–October 2, 2015, 374–380. doi:10.1109/IROS.2015.7353400
- Liu, Y., Wensing, P. M., Schmedeler, J. P., and Orin, D. E. (2016). Terrain-Blind Humanoid Walking Based on a 3-d Actuated Dual-Slip Model. *IEEE Robot. Autom. Lett.* 1, 1073–1080. doi:10.1109/lra.2016.2530160
- Mordatch, I., De Lasa, M., and Hertzmann, A. (2010). “Robust Physics-Based Locomotion Using Low-Dimensional Planning,” in ACM SIGGRAPH 2010 papers, 1–8. doi:10.1145/1833349.1778808
- Orin, D. E., Goswami, A., and Lee, S.-H. (2013). Centroidal Dynamics of a Humanoid Robot. *Auton. Robot* 35, 161–176. doi:10.1007/s10514-013-9341-4
- Raibert, M. H. (1986). *Legged Robots that Balance*. Cambridge, MA: MIT press.
- [Dataset] Robotics, A. (2017). *Cassie - Next Generation Robot*.
- Wang, K., Marsh, D., Saputra, R. P., Chappell, D., Jiang, Z., Raut, A., et al. (2020). “Design and Control of SLIDER: An Ultra-Lightweight, Knee-Less, Low-Cost Bipedal Walking Robot,” in Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS 2020), Las Vegas, USA, 24 Oct.-24 Jan. 2021. doi:10.1109/iros45743.2020.9341143
- Xin, S., Orsolino, R., and Tsagarakis, N. (2019). “Online Relative Footstep Optimization for Legged Robots Dynamic Walking Using Discrete-Time Model Predictive Control,” in Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS 2019), Macau, China, 3-8 Nov. 2019. doi:10.1109/iros40897.2019.8968028
- Xiong, X., and Ames, A. D. (2018). “Bipedal Hopping: Reduced-Order Model Embedding via Optimization-Based Control,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China, November 6–9, 2018 (IEEE), 3821–3828. doi:10.1109/iros.2018.8593547

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Wang, Fei and Kormushev. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.